Journal of Information Systems Education

Volume 31
Issue 1
Winter 2020

Exploring Which Agile Principles Students Internalize When Using a Kanban Process Methodology

Jeffrey Saltz and Robert Heckman

Recommended Citation: Saltz, J. & Heckman, R. (2020). Exploring Which Agile Principles Students Internalize When Using a Kanban Process Methodology. *Journal of Information Systems Education,* 31(1), 51-60.

Article Link: http://jise.org/Volume31/n1/JISEv31n1p51.html

Initial Submission:	10 January 2018
Accepted:	7 January 2019
Abstract Posted Online:	12 September 2019
Published:	3 March 2020

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Exploring Which Agile Principles Students Internalize When Using a Kanban Process Methodology

Jeffrey Saltz
Robert Heckman
School of Information Studies
Syracuse University
Syracuse, NY 13244, USA
jsaltz@syr.edu, rheckman@syr.edu

ABSTRACT

This paper reports on a case study of the Agile Kanban project methodology, which while growing in popularity, has had far less analysis on its usefulness in the classroom as compared to other frameworks such as Agile Scrum. Our study provides insight into why the Kanban methodology is useful by mapping student comments about the methodology to the twelve principles laid down in the Agile Manifesto. Our analysis identified two key agile principles that help to explain the value of Kanban. Specifically, we found that the students focused on self-organizing teams and reflection at regular intervals, and that these two principles led to improved team communication and coordination. Our findings are useful for those looking to use or define a process management methodology for student teams as well as others exploring the more general challenge of incorporating agile into the classroom.

Keywords: Agile, Project management, Kanban

1. INTRODUCTION

Project-based learning is widely used in post-secondary education and is often considered a key component of a student's education (Frame et al., 2015). Hence, educators have explored the use of project methodologies to help students better communicate and collaborate in their projects so as to help students overcome the challenges faced within their student project teams (e.g., Harding, 2017; Takai and Esterman, 2017). However, reviewing the use of agile process methodologies within computing courses led us to identify two key research gaps in how project management concepts are explored within project-based courses.

First, while there has been a research focus on the impact of teaching project methodologies to computer science students (e.g., Mahnic, 2012; Ding, Yousef, and Yue, 2017), or perhaps a bit more generally, to software engineering students, there has been less of a research focus for the information systems community of students. Since many of these information systems students will also participate in technology focused development after they graduate, it is important to teach these concepts to this broader population of students. In fact, it was recently noted that teaching agile is of growing importance within information systems education (Sharp and Lang, 2018) since the use of agile continues to gain prominence for information systems projects (Schmitz, 2018). One way to achieve this goal of teaching agile to information system students is to use a project management methodology in courses with a more diverse set of students as compared to the typical computer science software development capstone course.

Second, the focus in the classroom has typically been on the Agile Scrum framework (e.g., Mahnic, 2012; Ding, Yousef, and Yue, 2017). While Scrum has been shown to be useful, Kanban (Anderson, 2010) is an alternative framework that focuses on minimizing work-in-progress and has been shown to be even more useful for certain kinds of projects, such as those with uncertain outcomes. For example, one case study (Sjøberg, Johnsen, and Solberg, 2012) reported on a team that switched from Scrum to Kanban. That research noted that Kanban produced better outcomes than Scrum: after the team switched to Kanban, key metrics, such as lead-time, quality, and productivity all improved. In a different example, a controlled experiment within a classroom found that the use of Kanban yielded a higher quality result (Saltz and Heckman, 2018). Kanban's effectiveness has been attributed to several factors, such as its simplicity (Ikonen et at, 2011) and the fact that it allows the team to work in an agile manner without having to define how long a specific task might take, which is required with Agile Scrum (Saltz and Heckman, 2018).

Unfortunately, little has been done to understand which aspects of agility are most important when information system students use the Kanban process. In other words, our focus is not to explore if the Kanban methodology improves student team results (as others have noted), but rather, to explore which (if any) agile concepts students internalize while using Kanban, which might explain how the methodology helps student teams.

The twelve agile principles, described in section 2.1, are the foundational elements of agile practice and are still considered to deliver solid guidance (Williams, 2012). Hence, these principles are an appropriate lens to explore why an agile

methodology is useful. By exploring the Agile Kanban process via the lens of the agile principles, we aim to uncover the key drivers of the benefits achieved when students use the methodology. Hence, this research focuses on the following research question: Which, if any, agile principles do information system students most readily think about when using a Kanban-based project management methodology?

Understanding the most pertinent agile principles would be useful to enable instructors to explore enhancements to their current project-based learning process that could reinforce other agile principles. To address our research question, we report on a case study that focuses on a project within a broad-based introduction to data science course, but should be applicable to other information system focused student projects.

The rest of this paper first provides some background context. Then, Section 3 explains our research methodology. Section 4 notes our findings, and Section 5 discusses these findings. Finally, Section 6 presents our conclusion, including limitations and potential next steps.

2. PREVIOUS WORK

We first provide a brief overview of the 12 agile principles defined in the Agile Manifesto and then the Kanban approach to project management. We next discuss some of the challenges that faculty face when guiding student teams. This is followed by a review of the current state of data science in the classroom. We conclude this section by outlining the opportunities identified.

2.1 Agile Manifesto

In 2001, the Agile Manifesto was developed, which defined 12 basic principles to use when following an agile project management approach (Beck et al., 2001; Swanberg, 2018). As previously noted, these principles, described in Table 1, are the foundational elements of agile practice and are considered to deliver solid guidance to project teams.

Principle	Context/Comments
1. *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*	Do not lose sight of the fact that the goal of the project is to enable an end user to solve a problem or do their jobs better (which is different than just satisfying some initial requirements).
2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*	Don't be afraid to make changes. One doesn't need to wait for the next system to be built or a system redesign.
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*	Incrementally deliver a project, in addition, a project does not need to have 100% of the requirements known up front, before the project can start. Focus on creating the system, not planning on creating the system.
4. *Business people and developers must work together daily throughout the project.*	Co-location between management and developers can be helpful. The key is that the two sides better understand each other's perspectives, which can lead to better decision making and more productive work.
5. *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*	Teams should be self-directed and self-reliant (and hence micromanagement is not needed) and also make sure to provide the support and environment the team needs to get the job done.
6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*	When teams work together under the same roof, it's much easier to ask questions, make suggestions, and communicate.
7. *Working software is the primary measure of progress.*	This re-enforces the key focus on the importance of a working system, because if it's not working correctly, it can't be useful.
8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*	The key is to avoid burnout, which can be reduced by doing short bursts of work. This is important because excessive overtime cannot continue indefinitely without impacting the quality of the system.
9. *Continuous attention to technical excellence and good design enhances agility.*	Just as with a working system, the team should not wait to clean up redundant or confusing code. Doing this later often means never.
10. *Simplicity, the art of maximizing the amount of work not done, is essential.*	In other words, try to avoid doing things that don't matter.
11. *The best architectures, requirements, and designs emerge from self-organizing teams.*	The team should collectively be able to set its own direction, and not wait to be told what needs to be done -- they attack problems, clear obstacles, and find solutions.
12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.*	The team should be encouraged to identify process improvements, so if there is a better way of moving a project forward, the team should be empowered to implement those improvements.

Table 1. Agile Principles (Beck et al., 2001)

2.2 Kanban Project Management

Kanban was originally created for lean manufacturing (where Toyota line-workers used cards on a physical board to track progress and issues). In general, when using Kanban, one can think of the methodology as a process to see and manage the team's work pipeline. The following three key Kanban principles (Anderson, 2010) provide some context for how teams can execute the Kanban process:

- *Visualize the workflow* – Split the work into pieces; write each item on a card and put the card on the wall (either a physical wall, or more commonly today, a computer-based virtual wall). Use the named columns on the wall to illustrate where each item is in the workflow. By creating a visual model of the work and workflow, one can observe the flow of work moving through the Kanban system. Making the work visible is believed to lead to increased communication and collaboration.
- *Limit Work-in-Progress (WIP)* – This is typically done by having an explicit focus on WIP as well as assigning explicit limits to how many items may be in progress at each workflow state (column on the board). By limiting how much unfinished work is in process, the team can reduce the time it takes an item to travel through the Kanban pipeline. This can avoid problems caused by task switching and also provides agility by enabling new incoming tasks to be effectively prioritized.
- *Focus on Flow* – By using WIP limits and developing team-driven policies, the team can smooth the flow of work and make sure the team is focused on getting work completed.

There is growing research demonstrating the benefits when teams use a Kanban project management approach. For example, it has been empirically shown that Kanban provides increased motivation and project activity control (Ikonen et al., 2011). In a related finding, Sjøberg, Johnsen, and Solberg (2012) document a case study in which they gathered data from more than 12,000 work items collected over three years and found that, for software development, Kanban was more effective than Scrum. Anderson et al. (2012) did process modeling and simulation to explore the impact of Kanban on software development and found that the use of Kanban greatly improved team performance. In addition, Tripathi et al. (2015) explored the use of Kanban on larger software projects and found that setting WIP limits and visualizing product backlogs provided viable solutions for overcoming the challenges typically found in large-scale distributed projects when using Scrum, such as hierarchical requirements, large team size, and managing workflow.

A more recent study statistically compared the effectiveness of the Scrum and Kanban methods in terms of their effects on the project management factors for software development projects (Lei et al., 2017) and found that both Scrum and Kanban lead to the development of successful projects, but that the Kanban method was better in terms of managing and coordinating a project schedule. However, there has been no work examining if these types of teams achieved the key agile principles defined in the agile manifesto. In fact, Shafiq and Inayat (2017) noted that minimal work has been done on exploring the communication within Kanban teams and suggest a roadmap to study communication patterns of Kanban teams.

2.3 Guiding Student Teams

Student project teams have long been recognized as providing value by better preparing students for the reality of life after school and assisting students with critical-thinking and problem-solving skills (Frame et al., 2015). More generally, it has been noted that team-based learning results in deep learning by combining doing and thinking in practice (Koretsky, 2014). With respect to agile, project-based learning has been shown to enable students to acquire hands-on experience in coping with uncertain environments (Taipalus, Seppänen, and Pirhonen, 2018).

However, it has also long been recognized that there can be problems when using student teams. Such problems include free-riding or social loafing (Harding, 2017) and student attitude problems (Wolfe, 2008). In general, achieving a full contribution by all team members has been a persistent challenge that has not yet been mitigated (Takai and Esterman, 2017). To overcome these problems, much of the advice to faculty has focused on the idea of educating team members in the fundamentals of team dynamics and communication (Lam, 2015), creating effective student peer evaluations (Jassawalla and Sashittal, 2017), and improving the process of assigning students to teams (Harding, 2017).

Espinosa et al. (2007) remind us that the classic organization literature indicates that teams coordinate both *mechanistically* (using task programming mechanisms) and *organically* (achieved through team communication). As noted above, the traditional advice to faculty concerning instruction on teamwork and team dynamics is primarily intended to improve organic team coordination. Unfortunately, there are few studies that focus on a different facet of student preparation, namely, how mechanistic coordination can improve team coordination. Specifically, how a student's learning of a project coordination process methodology might affect the team. However, this is starting to change. With respect to the use of Kanban within project-based courses, Ahmad, Markkula, and Oivo (2014) report on a case study of students using the Kanban methodology and found that the students applied Kanban principles in their project work and perceived increasing success in the outcomes. It was also found that the majority of the students expressed positive views about Kanban in their project work and appreciated its value as part of their university education. Neyem et al. (2017) also report on the benefits of using a Kanban based methodology for capstone projects. Furthermore, as previously noted, Saltz and Heckman (2018) report on a controlled experiment exploring several different project methodologies for use on a project within a data science course and found that a Kanban-based methodology was more effective than other methodologies such as Agile Scrum.

2.4 Data Science in the Classroom

A data science course can address two key challenges when teaching agile. First, data science courses usually provide the opportunity for some programming, but do not require the extensive programming experience that a typical computer science course requires. This is important since teaching agile within the IS curriculum without focusing on coding is difficult and has not been frequently done (Cubric, 2013). Second, an

introduction to data science course typically also has a diverse range of student backgrounds (including business and information systems focused students) which helps to ensure that a broad base of students, beyond computer science and software development, will gain an appreciation of agile concepts. Another factor to consider is that, as shown by the increase in data science courses and programs (O'Neil, 2014), data science is a growing area of interest to students. Hence, data science is an interesting domain in which to explore the teaching of agile concepts.

However, perhaps because it is a new domain, beyond what was reported by Saltz and Heckman (2018), the only other reported research on the challenges that students might encounter when they are working on a data science project was from an earlier research effort from Saltz and Heckman (2016) where they discussed a project-focused data science course. However, that research was focused on the viability of using real world projects and did not address the question of how to best guide students through a project.

2.5 The Opportunity

Borrego et al. (2013) reviewed 104 articles describing computer science student team projects and noted that few of the articles discussed team effectiveness and concluded that there is a great opportunity to address this gap. This is particularly true for the broader information system student population and with respect to students using a Kanban framework. Thus, there appears to be an opportunity to more deeply explore why students perform effectively when using an Agile Kanban methodology.

3. METHODOLOGY

In this study, we explored the impact of using an agile process to guide information system students working on a group project via a case study. Merriam (1988) indicated that with case study research, it is important to have a bounded system that can be identified as the focus of the investigation. "A case study is an examination of a specific phenomenon such as a program, an event, a person, a process, an institution, or a social group" (Merriam, 1988, p. 9). This study examines the use of a Kanban process for the length of a data science course project.

3.1 Case Study Context / Environment

All students were in an introductory data science course that covered a range of concepts that were to be leveraged within the course project, such as visualization and machine learning techniques. Specifically, in this study, there were four sections of the course and two instructors, with each instructor teaching two of the sections. Students were put into teams to work on the semester-long project. The students were randomly assigned to sections and teams within the section, which were comprised of four to six students per team. All team members were from the same section. Since each section had between 22 and 24 students, each section had 4 student teams. Hence, there were 16 teams in our study (4 teams in the 4 sections).

There were four project updates during the semester where each team presented their Kanban board and discussed their results to date. The students did the project mainly outside of class, but for 5 of the weeks, the students had 30 minutes of work time within their class section. As part of this update, students discussed their thoughts on reprioritizing future tasks and goals for their next update, which were reflected in their Kanban board.

3.2 Case Study Project

The project started in week three of the semester and continued for the next 10 weeks. The project was done using the R programming language, a popular data science tool that is used in both industry and academia.

The project was positioned in a way that the teams were to analyze a large data set of customer survey responses for a client. The dataset was a modified version of a real dataset of survey responses. Hence, the data was not real, but was representative of the actual challenges one might face in executing a data science project. To do the analysis, the student teams were required to leverage many typical data science techniques, such as descriptive statistics, machine learning algorithms, and geographic information analysis.

The requirements for the project were high-level, and students were instructed to "help the management team understand the customer surveys and what drives customer satisfaction." Hence, each student team had to refine their goals (requirements) as they incrementally understood the data and what might be possible in terms of actionable insight generated via data analytics.

3.3 Agile Kanban Process Description

To help the students use the Kanban process, the students received an explanation of the process. The content of that discussion covered the key concepts reviewed in section 2.2 of the paper. The explanation took an hour of class time (including student Q&A) and covered the key concepts of Kanban: visualizing the workflow, limiting work-in-progress, and focusing on the flow of the work. In addition, the possible phases of a task (i.e., the columns on a Kanban board) were also discussed. However, the teams were given the freedom to define the columns on their board that they thought were most useful.

From a task perspective, each team was asked to define what they wanted to investigate (i.e., high level tasks, such as linking weather data to our previously collected data). These ideas were all listed (in a prioritized order) in their 'to do' column. The prioritization of tasks to be done was defined by the student team, and the students could change the priority of tasks at any time as the team thought was most appropriate. So, for example, insight gained from one completed task might impact the team's view on the priority of future tasks (that were in their 'to do' column). Then, as space permitted (based on the number of allowed simultaneous tasks at each step), a task was permitted to flow to the next column on the board. In other words, when a task was completed within a column, that task got moved to the next column and so on across the board until the task is completed. As the board allows (based on the work-in-progress limits), new tasks can be started. To help define and track work, the teams used trello (www.trello.com), which is a web-based tool for visualizing a board. Each team also decided on the size of the tasks (chunks of work) to be done. However, it was explained to the teams that the smaller and more detailed the task, the easier it would be for the team to understand potential bottlenecks.

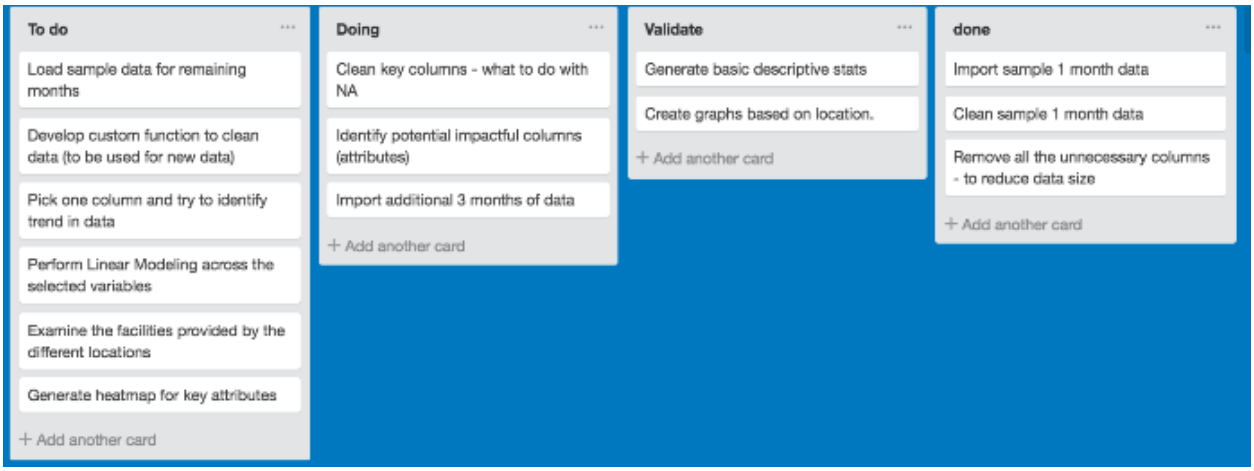


Figure 1. An Example Kanban Board

An example Kanban board is shown in Figure 1. Note that some of the tasks could have been improved (e.g., generate heatmap for key attributes – which is a very broad task), but this board is representative of what a team's board might look like in terms of the phases for a task (i.e., the columns) and the type of tasks defined by the team.

Throughout the semester, each team received feedback on their use of the methodology from their section instructor as well as from a TA for the course. Specifically, during each of the project checkpoints, the team's trello boards were reviewed by the instructor and feedback was provided on the tasks defined on the board as well as on the progress of the project. This enabled the students to refine their use of the methodology throughout the semester. It was via this feedback that the teams also received feedback with respect to the size of their tasks as well as how the students were prioritizing the different possible tasks (which was noted by exploring which tasks were in the 'to do' column versus tasks that were actually being worked on by the team).

Hence, the process to do a data science project could be thought of as a pipeline with requests entering one end and improved data insight coming out the other end. The team worked through the project pipeline throughout the project with no defined schedule, but rather, the team focused on ensuring that there was not a lot of time spent on an effort that did not complete (better to get a fewer number of tasks all the way through the pipeline).

3.4 Data Collection

Data for this study was obtained via open-ended survey questions to students. The elicitation stimuli for the students (i.e., the open-ended survey questions to the students) were purposely neutral regarding agile principles. Students were simply asked, "What were the advantages and disadvantages of using the Kanban methodology in your project?"

Note that the Kanban boards (implemented using trello) store the changes over time and that a tool was developed to be able to visualize the board at any previous date. Hence, even though the students were able to (and did) update the boards on a regular basis, we were able to obtain information on when each task was created, worked on, and completed.

3.5 Data Analysis

We explored the students' understanding of key agile concepts by mapping the student comments collected in our surveys to the 12 principles laid down in the Agile Manifesto which are still considered important for development teams and their projects (Williams, 2012) and were described in Section 2.1.

Specifically, we transformed the 12 agile principles into 12 independent codes used to determine if the students' open-ended responses to this neutral stimulus reflected an internalization of one or more of the agile principles. We divided each student response into sentences, and in some cases, divided student run-on sentences expressing several independent thoughts into more than one sentence/unit to be coded. Two independent coders evaluated each sentence/unit to determine if it expressed an understanding of one or more agile principles. The student responses contained 278 sentences to be coded (averaging 3.2 sentences per student). After training, the coders agreed on 87% of the coding decisions. Disagreements were discussed and agreed upon to create a final coded data set.

4. FINDINGS

In total, 93 students participated in the course, and 86 students responded to our survey (a 92% response rate). All participants were graduate students in an introduction to data science class. While most of the students were information systems students, 15 percent were business or public policy students. Eighty five percent of the students had previous IT experience, and approximately 60 percent of the students had previous information technology related work experience. However, the class had a wide range of students with diverse undergraduate majors, such as information technology, engineering, and business, and only five of the students had a traditional computer science educational background.

Of the 278 sentences coded, 204 (73%) showed evidence of one or more internalized agile concepts. Of the 86 student respondents, 83 (97%) showed evidence of internalizing at least one agile concept. Thus, when presented with a neutral elicitation stimulus that did not explicitly mention agile principles, virtually all students demonstrated internalization of at least some agile concepts. However, our focus was to

understand which concepts were most top of mind, which is shown in Table 2.

Specifically, Table 2 shows the percentage of students demonstrating internalization of each of the 12 agile principles and the number of coded sentences in which each principle was mentioned. Two principles (11 and 12) were readily elicited by a majority of students; six principles were elicited by a sizable number of students (1, 2, 3, 4, 8, and 10); and four principles were not elicited by any students (5, 6, 7, and 9). The following paragraphs discuss these findings in more detail.

Agile Manifesto Principle	Students Mentioning this Principle	Sentences Mentioning this Principle
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	**13** (15% of students)	**17** (6% of sentences)
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	**20** (23% of students)	**22** (8% of sentences)
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	**13** (15% of students)	**17** (6% of sentences)
4. Business people and developers must work together daily throughout the project.	**10** (12% of students)	**12** (4% of sentences)
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	**0**	**0**
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	**0**	**0**
7. Working software is the primary measure of progress.	**0**	**0**
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	**17** (20% of students)	**20** (7% of sentences)
9. Continuous attention to technical excellence and good design enhances agility.	**0**	**0**
10. Simplicity - the art of maximizing the amount of work not done - is essential.	**23** (27% of students)	**33** (12% of sentences)
11. The best architectures, requirements, and designs emerge from self-organizing teams.	**55** (64% of students)	**64** (23% of sentences)
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	**67** (78% of students)	**109** (39% of sentences)

Table 2. Agile Manifesto Principles and Coding Results

4.1 Continuous Delivery of Small Chunks

Principle 1, satisfying customers through early and continuous delivery of valuable work, and principle 3, deliver working software frequently, are closely related. In their surveys, thirteen students explicitly articulated this foundational concept, such as:

> It helps in delivering small portions of the bigger deliverable and enables a flexible response to any changes.
>
> It improves the delivery flow by promoting small, continuous changes in the system.
>
> Kanban supported a constant stream of independent improvements and optimizations.
>
> It also records the person's name who changes it, which gives a record of who is making what changes, which is crucial for agile environments considering the short project durations and how result oriented it is.

4.2 Flexibility

Twenty students recognized that Kanban promoted principle 2, which welcomes changing requirements, and shows that they understood the need to flexibly reprioritize what needs to be done as requirements change. The Kanban boards demonstrated this flexibility by having new tasks added to the 'to do' column, and that those newer tasks were sometimes executed prior to older 'to do' tasks. Below are some specific examples of student thoughts on flexibility:

> We were able to update the board on an iterative basis with new tasks … with each update.
>
> It freed us up to shift priorities as needed.
>
> It provides a great flexibility in its iterations.
>
> It helps in delivering small portions of the bigger deliverable and [provides a] flexible response to any changes [deemed important by the team].

4.3 Frequent Contact between Developers and Clients

Principle 4 describes the importance of frequent (even daily) communication and synchronization between developers and their clients. While students did not directly express the

importance of daily contact with clients, 10 students recognized the importance of tightly synchronizing their work with their professor, a client of sorts in this context. They articulated the benefit of the Kanban board in doing so. Several also recognized the value of real-time updates (a proxy for daily contact) for the synchronization process. Below are some example student comments:

> It was a platform for the professors as well as my group members to coordinate and communicate.
>
> Advantage: Real Time Updates.
>
> [The process and the trello tool] lets the professor look inside the progress of the project.
>
> The main advantage is that Trello and Kanban provide a platform that keeps the whole team [as well as the] professors in sync.

4.4 Building Projects around Motivated Individuals

Principle 5 was not addressed by student comments. This might not have been mentioned since the students were assigned to teams, and the motivation was driven by the project grade which was a significant aspect (35%) of the course grade.

4.5 Meet Face-to-Face

Principle 6 was not addressed by student comments. This might not have been top of mind since the students typically met once a week, outside of class, in a face-to-face manner. However, meeting face-to-face outside of class was not required, and many teams had synchronous and asynchronous team discussions in a non-face-to-face context.

4.6 Focus on Working Software

Principle 7 was also not mentioned by the students. It is possible that this was due to the fact that data science projects are different than software projects, and students thought more about continuous delivery of small chunks (principles 1 and 3 that were previously noted).

4.7 Sustainable Development, Pace and Flow

Seventeen students recognized principle 8, which focuses on sustainable development, in that the strong visualization of the Kanban board contributes to the steady, sustainable flow of work that is a fundamental agile characteristic. Students demonstrated their understanding of this concept by statements such as:

> [The process] introduces transparency into the backlog of work items and encourages focused and open communication.
>
> It is a strong interface where we can systematically schedule our upcoming tasks and be aware of the current status of our project work.
>
> It improves the delivery flow by promoting small, continuous changes in the system.

4.8 Continuous Technical Improvement

Principle 9 was not mentioned by the students, perhaps due to the nature of the course and the fact that students perceived technical learning as a key part of the course.

4.9 Simplicity and Minimize Work-in-Progress

Principle 10, which focuses on simplicity, hints at minimizing work in progress. Twenty-three students understood the concept and benefits of minimizing WIP, as demonstrated by statements such as:

> It helped in focusing on controlling the amount of work the team had in progress at any point in time.
>
> [It helped to enable a] reduction of wasted work, increased productivity, increased efficiency.
>
> It is easy to understand and reduces waste from the process.
>
> [The process] increases [the] ability of members to focus on work and [helps cause a] reduction of wasted time.
>
> Kanban involves WIP management criteria. Thus by managing WIP and monitoring WIP we can optimize the flow of work items.
>
> The team was able to adjust the work in progress level dynamically to avoid being idle.

4.10 Self-Organization

A majority of students (55) mentioned, sometimes implicitly, Principle 11, which focuses on the need for self-organization in teams. The comments focused on division of labor, effective coordination, individual accountability, and critical practices for self-organizing teams. Below are some example student comments:

> Kanban helped to assign and regularize project activities among the team members.
>
> [Kanban] empowers a team to self-manage visual processes and workflows.
>
> We used [Kanban] to make sure our project is in progress and each of us is contributing something.
>
> [Kanban] improves communication between yourself and others on your team. [It] inspires team collaboration.
>
> Agile Kanban boards were useful for the coordination between the team where our tasks were segregated and the progression towards the completeness was monitored.
>
> The main advantage was being able to keep track of each task that needed to be accomplished, and who the tasks were assigned to.

4.11 Process Reflection

Principle 12, which focused on team reflection, was the most commonly recalled agile concept, mentioned by 67 students. The student comments paint a picture of teams that reflected on

their internal process and looked for ways to improve them, and the comments included:

> The decision for when to pause the workflow can be triggered by date, by milestone, or by other metrics agreed to by the team before the work begins.
>
> Kanban helped maximize our productive work and helped us visualize a better scenario of our completed work, ongoing and incomplete work.
>
> Kanban is a practice, so teams can leverage its principles in their everyday work instead of having to stop what they are doing to focus on a new improvement initiative.
>
> [Because] Kanban is event-driven instead of timeboxed, it helps in tracking [our] project.
>
> Keeping a check on the health of project is easier as 'to-do' tasks can be compared to tasks Completed.

5. DISCUSSION

By using an Agile Kanban approach, the student teams focused on a few tasks at a time (thus limiting WIP). As teams finished tasks, they learned from the results of those tasks to help determine what is the most appropriate next task to start. In other words, based on results of the completed tasks, students understood that they could do a quick redirection to a new task based on the team's newly updated knowledge. This required communication and coordination in what tasks to do next.

Specifically, the two key agile principles most frequently mentioned by the students were self-organizing teams and reflection at regular intervals. At a higher level, these two concepts were likely merged within the students' minds in that the process enabled/required the teams to have improved coordination and communication. One student directly noted this concept by stating that "using the Agile Kanban methodology enabled everyone to better understand our project status and also improved the team coordination." Hence, we find that the reason the process was useful was that students gained an improved ability to coordinate and communicate. This thought process was embodied by comments of other students, including comments such as "it was easier to understand and distribute tasks" and "(Kanban) improves communication between yourself and others on your team. It inspires team collaboration" and "It is a method which helps maintain coordination between the team and helps us understand the status of the project," or more simply, "It helped the team to be on the same level throughout the project."

We posit that the use of Kanban in the course project led students to internalize these key agile principles in two ways. First, it helped them structure shared knowledge of the task and of the team. Second, it provided a shared information artifact that functioned as a mediator of distributed social cognition, facilitating shared memory, coordination, and communication.

5.1 Visible Team Knowledge

We believe that making the work visible, especially seeing work-in-progress bottlenecks, led to increased communication and collaboration. This was because making the work visible helped the teams create improved team knowledge, knowledge that is shared across the team, a form of a shared mental model. In several studies, team knowledge has been used as a framework to understand the factors contributing to the success of a process. Espinosa et al. (2007) found that team knowledge was important in overcoming three critical types of dependencies: technical, temporal, and process. They identified two types of team knowledge that were important in coordinating the work of software teams: (a) shared knowledge of the task and (b) shared knowledge of the team. If a team has a high level of shared team knowledge, then team members can more effectively self-organize to prioritize the work that needs to be done. This is especially important with data science projects where there is significant ambiguity in the tasks that should be completed as well as the duration of those tasks.

5.2 Information Artifacts

We can also turn to research on information artifacts. Such artifacts represent shared team knowledge and are often used to coordinate team actions. There have been varying approaches to studying the way shared information artifacts impact teams. One stream of research explores how shared artifacts act as mediators of shared distributed cognition. Rambusch, Susi, and Ziembke (2004, p. 1113) argue that cognition should not be exclusively regarded as an internal, individual process, but rather that "cognition cannot be separated from the social and material environment in which people live and act, and that in many cases cognition is distributed among individuals and environmental properties."

In this stream of research, shared information artifacts, such as the visual Kanban board, function as mediators of distributed social cognition. When a student in this study commented that "everyone has to agree to use it for [the Kanban board] to be useful," the student reinforced the idea that team context (norms, attitudes, expectations) are important determinants of how effective a process methodology is likely to be.

6. CONCLUSION

6.1 Limitations and Future Research

One limitation in this research is that the approach was only evaluated in a data science class. Therefore, one possible avenue for future research is to do additional studies to attempt to replicate and expand these findings in other types of project-based courses. In addition, one could evaluate the outcomes using both Scrum and Kanban frameworks, so as to be able to compare Scrum versus the Kanban process that was used in this study. Another limitation was that this study focused on graduate students; it is possible that exploring these concepts with undergraduate students might yield different results.

While not a key focus of our case study, we did note the impact of using this methodology on the workload of the faculty. Our observations suggest that, as compared to not using the methodology, the students required additional faculty support, especially during the startup of the project. For example, the instructors had to work with students to help them understand the importance of task granularity. Hence, in the future, it would be helpful for instructors to be able to more efficiently ensure proactive guidance on size of tasks by being able to quickly review the Kanban boards in a streamlined manner.

Of course, not all the comments with respect to the methodology were positive in nature. For example, several students complained about the lack of attention to time estimates, lack of integration with the work and the actual R programming code, and speculation that Kanban might not be ideal for large and complex projects. This suggests some students wanted to use project management techniques that they had previously used. Future case studies could explore providing additional explanation of why Kanban does not require some of these more traditional deliverables. The explanation of Kanban could also include more discussion on the benefits of using a Kanban approach (e.g., for student teams, task estimation is difficult and error prone, and hence, a process that does not require these deliverables has some inherent advantages). Student teams that were provided this additional explanation could be evaluated to see if it addressed the student complaints, or if there are issues with the methodology that need to be addressed via a refined methodology.

6.2 Summary

This study explored which agile concepts a broad range of information system students, using the Agile Kanban framework within a data science course, most readily identified. Specifically, we found that two principles were recalled by the majority of students surveyed. These two principles focus on team reflection (12. At regular intervals the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly) and self-organization (11. The best architectures, requirements, and designs emerge from self-organizing teams). Thus, we addressed our research question (which, if any, agile principles do information system students most readily think about when using a Kanban-based project management methodology). Understanding which agile principles were most readily recalled can help provide context as to why the Agile Kanban process was useful.

7. REFERENCES

Ahmad, M. O., Markkula, J., & Oivo, M. (2014). Kanban for Software Engineering Teaching in Software Factory Learning Environment. *World Transactions on Engineering and Technology Education*, 12(3), 338-343.

Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. *Blue Hole Press*.

Anderson, D. J., Concas, G., Lunesu, M. I., Marchesi, M., & Zhang, H. (2012). A Comparative Study of Scrum and Kanban Approaches on a Real Case Study using Simulation. In *International Conference on Agile Software Development*, 123-137.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Humt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thom, D. (2001). *Manifesto for Agile Software Develo*pment. http://agilemanifesto.org/.

Borrego, M., Karlin, J., McNair, L., & Beddoes, K. (2013). Team Effectiveness Theory from Industrial and Organizational Psychology Applied to Engineering Student Project Teams: A Research Review. *Journal of Engineering Education*, 102(4), 472-512.

Cubric, M. (2013). An Agile Method for Teaching Agile in Business Schools. *The International Journal of Management Education*, 11(3), 119-131.

Ding, D., Yousef, M., & Yue, X. (2017). A Case Study for Teaching Students Agile and Scrum in Capstone Course. *Journal of Computing Sciences in Colleges*, 32(5), 95-101.

Espinosa, A., Slaughter, S., Kraut, R., & Herbsleb, J. (2007). Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems* 24(1), 135-169.

Frame, T. R., Cailor, S. M., Gryka, R. J., Chen, A. M., Kiersma, M. E., & Sheppard, L. (2015). Student Perceptions of Team-Based Learning vs Traditional Lecture-Based Learning. *American Journal of Pharmaceutical Education*, *79*(4), 51.

Harding, L. M. (2017). Students of a Feather "Flocked" Together: A Group Assignment Method for Reducing Free-Riding and Improving Group and Individual Learning Outcomes. *Journal of Marketing Education*, 40(2), 117-127.

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation. In *16th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*.

Jassawalla, A. & Sashittal, H. (2017). What Students Think and Do in Classroom Teams when Peer Evaluations are Highly Consequential: A Two-Stage Study. *Decision Sciences Journal of Innovation Education*, 15(2), 219-247.

Koretsky, M. D., Gilbuena, D. M., Nolen, S. B., Tierney, G., & Volet, S. E. (2014). Productively Engaging Student Teams in Engineering: The Interplay between Doing and Thinking. In *Frontiers in Education Conference (FIE),* 1-8.

Lam, C. (2015). The Role of Communication and Cohesion in Reducing Social Loafing in Group Projects. *Business and Professional Communication Quarterly*, 78(4), 454-475.

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A Statistical Analysis of the Effects of Scrum and Kanban on Software Development Projects. *Robotics and Computer-Integrated Manufacturing,* 43, 59-67.

Mahnic, V. (2012). A Capstone Course on Agile Software Development using Scrum. *IEEE Transactions on Education,* 55(1), 99-106.

Merriam, S. (1998). *Qualitative Research and Case Study Applications in Education*. San Francisco, CA: Jossey-Bass Publishers.

Neyem, A., Diaz-Mosquera, J., Munoz-Gama, J., & Navon, J. (2017). Understanding Student Interactions in Capstone Courses to Improve Learning Experiences. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 423-428.

O'Neil, M. (2014). As Data Proliferate, So do Data-Related Graduate Programs. *The Chronicle of Higher Education*, February.

Rambusch, J., Susi, T., & Ziemke. T. (2004). Artefacts as Mediators of Distributed Social Cognition: A Case Study. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 26, 1113-1118.

Saltz, J. & Heckman. R. (2016). Big Data Science Education: A Case Study of a Project-Focused Introductory Course. *Themes in Science and Technology Education*, 8(2), 85-94.

Saltz, J. & Heckman, R. (2018). A Scalable Methodology to Guide Student Teams Executing Computing Projects. *ACM Transactions on Computing Education*, 18(2), Article 9.

Schmitz, K. (2018). A Three Cohort Study of Role-Play Instruction for Agile Project Management. *Journal of Information Systems Education*, 29(2), 93-103.

Shafiq, S. & Inayat, I. (2017). Towards Studying the Communication Patterns of Kanban Teams: A Research Design. In *IEEE 25th International Requirements Engineering Conference Workshops*, 303-306.

Sharp, J. H. & Lang, G. (2018). Agile in Teaching and Learning: Conceptual Framework and Research Agenda. *Journal of Information Systems Education*, 29(2), 45-51.

Sjøberg, D. I., Johnsen, A., & Solberg, J. (2012). Quantifying the Effect of using Kanban versus Scrum: A Case Study. *IEEE Software*, 29(5), 47-53.

Swanberg, K. (2018). *The 12 Basic Principles of Agile Project Management*, Retrieved on November 25, 2018, from https://blog.hubspot.com/agency/basic-principles-agile-project-management.

Takai, S. & Esterman, M. (2017). Towards a Better Design Team Formation: A Review of Team Effectiveness Models and Possible Measurements of Design-Team Inputs, Processes, and Outputs. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Taipalus, T., Seppänen, V., & Pirhonen, M. (2018). Coping with Uncertainty in an Agile Systems Development Course. *Journal of Information Systems Education*, 29(2), 117-126.

Tripathi, N., Rodríguez, P., Ahmad, M. O., & Oivo, M. (2015). Scaling Kanban for Software Development in a Multisite Organization: Challenges and Potential Solutions. In *International Conference on Agile Software Development*, 178-190.

Williams, L. (2012). What Agile Teams Think of Agile Principles. *Communications of the ACM*, 55(4), 71-76.

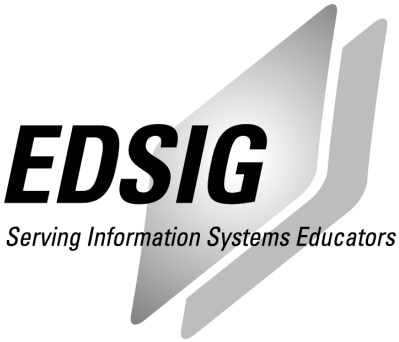
Wolfe. A. (2008). Student Attitudes toward Team Projects. *Proceedings of the Marketing Management Association*, 119-126.

AUTHOR BIOGRAPHIES

Jeffrey Saltz is an associate professor at Syracuse University, where he leads their graduate applied data science program. His research focuses on data science project management and agile development. Prior to joining Syracuse, he worked as the head of technology for risk and authorizations for Chase Credit Card, Vice President of computational technology for JP Morgan, and Chief Technology Officer at Goldman Sachs/Goldman Sachs Ventures. He started his career as a technology leader with Digital Equipment Corp and holds a B.S. degree in computer science from Cornell University, an M.B.A. from the Wharton School at the University of Pennsylvania, and a Ph.D. in information systems from the New Jersey Institute of Technology.

Robert Heckman is Professor Emeritus at Syracuse University. He has previously served as graduate program director, associate dean for academic affairs, and senior associate dean in the school of information studies at Syracuse. His research interests include design of work-based learning experiences, learning strategies for information professionals, self-directed learning, and collaboration in virtual communities and teams. Previously, he had more than 20 years of experience in the information services industry as a senior manager of data processing operations, systems development, and information systems marketing. He received his Ph.D. in information systems from the Katz Graduate School of Business at the University of Pittsburgh.

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 2574-3872